Drishti: An Ultra-Low Cost Visual-Aural Assistive Technology for the Visually Impaired

ABSTRACT

We describe a Visual-Aural aid to support the visually impaired by providing them with aural feedback in form of natural speech representing their visual environment in a relatively inexpensive way. The scene is captured by an ordinary webcam and we use methods to detect, recognize and track objects, faces and text in 2D images and translate them into speech which the visually impaired individual hears via earphones. This paper describes the architecture, implementation and the operation of the entire system Drishti, Sanskrit for faculty of sight.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation]**: Multimedia Information Systems – *audio input/output* ; H.5.2 [**Information Interfaces and Presentation]**: User Interfaces – *natural language* ; I.2.1 [**Artificial Intelligence]**: Application and Expert Systems – *medicine and science, natural language interfaces* ; I.2.7 [**Artificial Intelligence]**: Natural Language Processing – *speech recognition and synthesis, text analysis* ; I.2.10 [**Artificial Intelligence]**: Vision and Scene Understanding – *shape, texture* ; I.4.8 [**Image Processing and Computer Vision]**: Scene Analysis – *object* recognition ; I.5.4 [**Pattern Recognition]**: Applications – *computer vision, text processing* ; K.4.2 [**Computers and Society]**: Social Issues – *assistive technologies for persons with disabilities.*

General Terms

Algorithms, Measurement, Design, Experimentation, Human Factors

Keywords

Aid for Blind, Aid for Visually Impaired, Assistive Technology, Computer Vision, Face Recognition, Object Recognition, Text Recognition, Visually Impaired, Vision for Blind.

1. INTRODUCTION

As per WHO October 2013 report [1], there are estimated 285 million people who are visually impaired worldwide, of which 39 million are totally blind. 90% of them live in developing countries (India accounts for biggest share), majority are above the age of 50 and do not enjoy equally opportunities due to their handicap, relegating them aside and rendering them economically disadvantaged. Despite technological advancements in computing, smart phones, sensors, almost all rely on age old white cane.

The most popular methods of coping with the disabilities, besides the white cane are guide dogs, but these are not popular in developing countries due to high initial and ongoing costs associated with keeping a dog. Several innovative ideas have been developed like ultrasonic sensor attached cane [2], Le Chal [3], a haptic shoe for the blind, SeeMore [4], The Haptic Cane but these are all Electronic Travel Aids (ETA) and don't address a visually impaired individual's need to "see" her environment and be able to perform tasks that could perhaps provide employment opportunities.

Millions of dollars are being spent in developing technologies like 3D sensors [5] and associated systems but unfortunately being very expensive these are unlikely to benefit the disadvantaged in the developing world.

Our primary aim was to build and demonstrate an aid to provide aural feedback in the form of voice, in an inexpensive manner, using cheap and readily available peripherals like a webcam and earphones under $50, a computing device like a laptop or mobile phone and associated software to help improve their lives and employment opportunities.

2. PROBLEM SCOPE

In order to establish attainable goals, the first task was to identify scope and target user group. This was to ensure that a system can be developed in a reasonable time and end budget.

We first chose to focus on visually impaired individuals who could be assisted in an environment like an office or factory where they could perform finite tasks but their productivity and accuracy could be substantially improved giving them emotional confidence. The inexpensive hard-ware and computing device could be provided by the employer.

Once scope and target user group was identified, methods of decoding and parsing the image to convert them to audio for interpretation was to be achieved. This had to be done under the constraint of computing power on board. The software had to focus on decoding text and identify everyday objects and people.

3. DETAILED FRAMEWORK

The framework shown in Figure 1 is broken down into three software modules, each module serving a particular vision algorithm and a speech module running on Microsoft Windows

laptop to which a USB camera and a pair of earphones are attached via audio jack.

The test set up grabbed images in RGB format with resolution 640x480 pixels at 30 fps. These are piped to the three modules. If any module is busy the frame is discarded. The frames are concurrently processed by the three modules and then fed to the Natural Language Generation (NLG) module for constructing the scene and thereafter aural output.

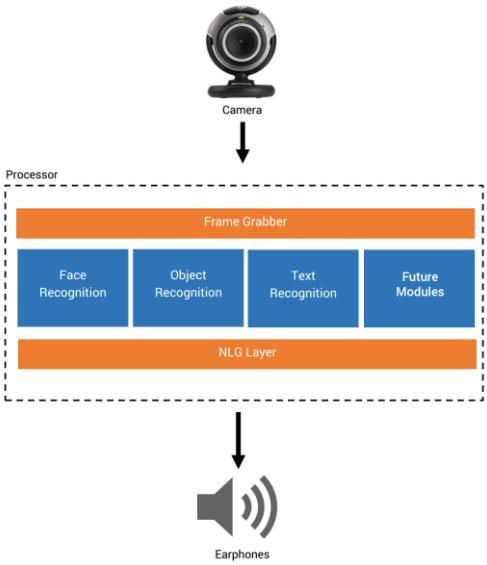

Figure 1. Drishti System Architecture.

3.1 Text Recognition

For the system to work effectively, the Text Recognition module was designed to read sign boards and text from a document or novel.

Conventional Optical Character Recognition (OCR) methods failed here as the image contains a large amount of clutter [6]. These methods expect a clear image with only text.

OCR was done by analyzing and classifying contours in our query image. We detected the edges of the image using Canny edge detection algorithm [7] and identify the contours. Since Canny edges return a good gradient map, printed text ends up as a contour, due to the high contrast.

Once the contours have been extracted, a Support Vector Machines (SVM) based classification was used to identify the letters. The classifier was fed with Histogram of Oriented Gradient (HOG) as feature vectors. We used alpha-numeric characters of a set of generic fonts as the training data for the classifier. A high threshold was set for detection confidence to reject contours which were not characters. Since we rotate the characters for training and de-skew the contours, this method is both rotation and scale invariant.

This is a major flaw for alphabets in the roman script. E.g. the system gets confused between ‘Z’ and ‘N’. To tackle this we assumed that the letters can’t be rotated more than a specific angle. We restricted the rotation of the training set.

Once the contours were correctly identified, we tried and formed words or sentences out of them. We clustered character contours using k-means approach and identified words.

To identify sentences we simply used the position of the words and gave priority to the words closer to the origin.

Figure 2. Query image for text recognition.

The advantage of this algorithm is that it can be retrained for any language and can identify words and sentences even when there is a lot of clutter in the scene.

Figure 2 represents a sample of image used for testing the approach.

Table 1. Table captions should be placed above the table

Algorithm	Execution Time (ms)	Recognized Text
Tesseract	540	iiiid iijdi
Contour Classification	54	FIRE EXIT

Table I gives us the performance of each OCR algorithm. While Tesseract is extremely powerful for accurate OCR of documents, contour classification worked better in our situation.

3.2 Face Recognition

The ability for a computer to recognize faces isn’t new and existing algorithms proved to be extremely fast to be used on a mobile device.

We used the Voila-Jones method [8] (HAAR Wavelets) to detect faces in an image and using the Principle Component Analysis (PCA) [9] we classified and labeled the faces.

We used three cascades for face detection for three different sides of the face. The system can deal with both front and side profiles and thus needs to be trained in three orientations as well.

The system was found to be robust against scale and orientation variances and functioned well irrespective of lighting conditions within normal acceptable range.

3.3 Object Recognition

Multiple attempts went into perfecting this part of the system. This was due to the lack of existing solutions to the problem that we had in hand.

Our first attempt was using a feature based classification system. We used feature detector and descriptors like Speeded up Robust Features (SURF) [10] and used a nearest-neighbor approach to match features. This was extremely accurate and robust, however the computation time scaled linearly as more objects were trained.

We then tried to implement a contour based algorithm similar to the one described earlier in Section 3.1. This approach used Canny edges based contours and an SVM based classifier extending our Text Recognition module to a more general target. With a brilliant execution time this had several drawbacks.

1. This only worked for objects with a nice contrast to its background, a camouflaging effect took place when similar shades of objects was present.
2. Classification was not so accurate as object features were not distinct as characters.
3. As every contour was classified, it became almost impossible using the confidence to reject false-positives.

This prompted us to look for a more generalized approach. The algorithm we utilized was Associative Video Memory (AVM) - Yeremeyev method [11]. AVM is based on multilevel decomposition of recognition matrices. As this method is not rotation invariant, we rotated the training set and fed it to the classifier. However the system was prone to return false positives even after thresholding the confidence values.

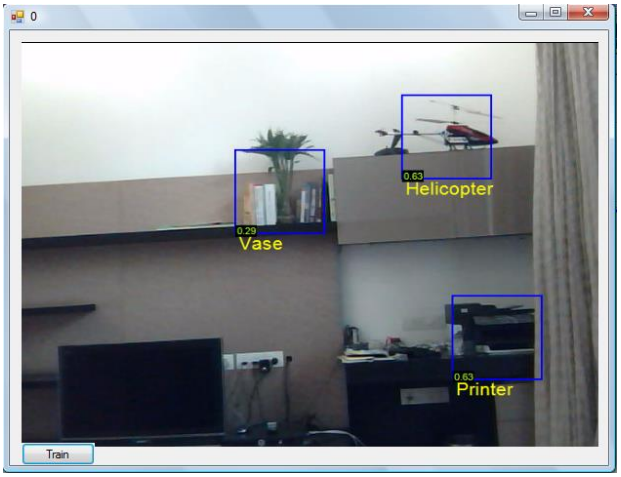

Figure 3. Results from the AVM and DNN algorithm. Confidence values are displayed at bottom left of the bounding box.

We used a Deep Neural Nets (DNN) [12] based approach to classify the object once detected by the AVM algorithm. This proved to be a computationally inexpensive way to correct the error.

Vote the confidence value for vase in Figure 3, which is below the threshold was still detected.

This approach had several advantages:

1. Low computation time, making it a real-time algorithm.
2. Didn't require images stored locally for training, training data is stored in the form of neural weights and labels.
3. Performed well when tracking, giving us relative position of the objects real-time.

3.4 Aural Output

Once the image was processed, we went about to convert labels and position into words. We studied sports commentary on a radio to understand how audio can be used to describe a scene.

The approach involved the use of Natural Language Generation (NLG) engine to produce a natural sentence that could be used by a text-to-speech engine to give an audio output.

However the above wasn't implemented, instead a simpler custom NLG module was used for the text-to-speech engine. This would take in processing data and frame sentences describing the relative position and label of the scene.

The scene depicted in Figure 3 is read out as "There is a vase on left of the helicopter which is above the printer".

We used open source text-to-speech engines to speak out the sentences framed.

4. EXPERIMENTAL SETUPs

The aim of the system was to deliver a low-cost solution for visually impaired people. As the main bottleneck for the cost lay in the hardware, we tried to minimize the cost here.

We designed a simple prototype using an ordinary USB webcam attached to a simple cap and a pair of earphones as shown in Figure 4.

Figure 4. Webcam attached to a cap.

A more minimal apparatus could have been designed but is out of the scope of this paper. Options like lapel camera could also be used or a much smaller web cams.

The hardware cost certainly less than $50 beyond a laptop making it ideal to be used in developing countries. The system weighed (even with the cam we had in our lab) approximately 200 grams as opposed to existing solutions requiring heavy 3D sensors mounted on helmets.

5. RESULTS and ANALYSIS

The system was trained with over 500 objects and performed well.

Text recognition performed with greater accuracy using $1/10^{th}$ of time when compared to using Tesseract. Using three cascades for representing three sides of a face improved recognition and did not require materially extra storage. Object recognition was fast and performed well when tracking.

Overall, the system coped with simultaneous feed of object, face and text and NLG was able to voice out all. Average frame processing time was in region of 110 ms.

One interesting use case was recognizing a watch. The system spoke out "Analogue Watch" when we trained it first but we felt that it would be more useful if the time was read out. We trained the system to recognize pattern of hour and minute hands in intervals of 15 minutes and the system read out time to the nearest 15 minutes.

We added a feature to the system for recognizing currency notes as part of user feedback and Drishti was able to speak out the denomination of the notes.

The software modules were compiled for Microsoft Windows operating system which is readily available in most user environments.

We tested the system with visually impaired subjects to help pack books and other articles in delivery boxes and it did serve the intended purpose of improving both speed and accuracy.

6. CONCLUSIONS

The most expensive and bulky device is the laptop and we plan to optimize and repurpose the software including removing extraneous libraries and making code much more compact to run on new generation of Snapdragon / ARM processors coming on mobile devices such as smartphones. We plan to shrink our hardware and use open-source forms of computation such as Raspberry Pi and Beagle Board XM from Texas Instruments for people without a mobile devices to bring costs down and making this more portable and adoption friendly.

The system needs to evolve further. The algorithms described and used are bound to improve and become faster over time. The implementation of the NLG engine is essential for correct interpretation. We plan to replace the laptop /computer with a mobile device. This is easy as the system is computationally inexpensive to run.

We also plan to continue to run some extensive usability tests and continue to improve the system via field trials.

The software is planned to be released back as open source.

Many visually impaired are missing a chance of employment opportunities and a dignified life. Our motivation was to develop a low cost and simple assistive technology to cater to this need and make it more accessible. We believe the system fulfils this purpose by applying an array of computer technologies, algorithms, simple hardware and principles of frugal engineering.

7. ACKNOWLEDGEMENTS

We wish to thank Eskay Charitable Trust, Dehradun, India for assisting us with field trials, all the visually impaired users who regularly visit their facility and provided us with valuable feedback and encouraged us to continue with this work.

8. REFERENCES

[1] World Health Organization, Media Centre, *Visual Impairment and Blindness.* http://www.who.int/mediacentre/factsheets/fs282/en/

[2] Kuchenbecker, K. J and Wang, Y. *HALO: Haptic Alerts for Lo Hanging Obstacles in White Canes.* IEEE Haptics Symposium 2012

[3] Sharma, A. *Le Chal: A Haptic Feedback Based Shoe for the Blind*. http://anirudh.me/2011/06/le-chal-a-haptic-feedback-based-shoe-for-the-blind/

[4] Gomez, D. and Saavedra, N. *SeeMore: A Haptic Cane*, http://reedlab.eng.usf.edu/photos/HapticsDemos2012/2012GomezSaavedra.pdf

[5] CASBliP Project., http://casblipdif.webs.upv.es/index.html

[6] Smith, R. *An Overview of the Tesseract OCR Engine*, http://tesseract-ocr.googlecode.com/svn/trunk/doc/tesseracticdar2007.pdf

[7] Canny, J. *A Computational Approach to Edge Detection*. IEEE Trans. PAMI 8, 6, pp. 679-698, 1992

[8] Jones, M and Viola, P. *Robust Real-Time Face Detection.* International Journal of Computer Vision, 57 (2), pp. 137–154, 2004

[9] Baek, K., Bartlett, M. A., Beveridge, J. R. and Draper, B. A. *Recognizing faces with PCA and ICA.* Computer Vision and Image Understanding, 91, 1-2, pp. 115-137, 2003

[10] Bay, H., Ess, A., Gool, L. V. and Tuytelaars, T. *SURF: Speeded Up Robust Features*. Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008

[11] Yeremeyev, D. V. *Associative Video Memory*. http://edv-detail.narod.ru/AVM_main.html

[12] Erhan, D., Szegedy, C. and Toshev, A. *Deep Neural Networks for Object Detection*. http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1210.pdf